

Process Mining Analysis of Conceptual Modeling Behavior of Novices - empirical study using JMermaid modeling and experimental logging environment

Abstract

Previous studies on learning challenges in the field of modeling focus on cognitive perspectives, such as model understanding, modeling language knowledge and perceptual properties of graphical notation by novice business analysts as major sources affecting **model quality**. In the educational context **outcome feedback** is usually applied to improve learning achievements. However, not many research publications have been written observing the characteristics of a **modeling process** itself that can be associated with better/worse learning outcomes, nor have any empirically validated results been reported on the observations of modeling activities in the educational context. This paper attempts to cover this gap for conceptual modeling. We analyze modeling behavior (conceptual modeling event data of 20 cases, 10.000 events in total) using experimental logging functionality of the JMermaid modeling tool and process mining techniques. The outcomes of the work include **modeling patterns** that are indicative for worse/better learning performance. The results contribute to (1) improving teaching guidance for conceptual modeling targeted at **process-oriented feedback**, (2) providing recommendations on the type of data that can be useful in observing a modeling behavior from the perspective of learning outcomes. In addition, the study provides first insights for learning analytics research in the domain of conceptual modeling.

Keywords: teaching/learning conceptual modeling, process-oriented feedback, conceptual modeling pattern, information systems education, process mining, learning data analytics

1. Introduction

Empirical studies show that more than half the errors that occur during systems development are requirements errors (Endres, 2003; Lauesen, 2001). Requirements errors are also the most common cause of failure of development projects (Moody, 2005; Schenk, Vitalari, & Davis, 1998). The success of the analysis of requirements depends heavily on models. Formalization of requirements through **models** enables quality control at a level that is impossible to reach with requirements articulated in natural language (Sikora, Bastian, & Pohl, 2011). With the growing importance of compliance between business strategy and ICT realizations as well as the emergence and evolution of Model Driven Engineering (MDE), **conceptual modeling** gains more relevance. Teaching conceptual modeling skills is however a challenging task. In their early careers novice modelers produce incomplete, inaccurate, ambiguous, and/or incorrect models (Schenk, et al., 1998). Errors occurring early in the systems analysis process are much more expensive and time-consuming to resolve when only detected later in the engineering process than those that may occur at any other time in systems engineering (Schenk, et al., 1998). Studies on *learning quality improvements* indicate a self-regulative approach as major source of impact on learning outcomes which in turn is closely intertwined with **feedback** research (Nicol & Macfarlane-Dick, 2006; Zimmerman, 2008), i.e. for all self-regulative activities, external feedback is considered as an inherent catalyst (Barber, Bagsby, Grawitch, & Buerck, 2011; Butler & Winne, 1995). As proposed by the constructivist approach (Hadjerrouit, 2005) the method of **dialogue** is the most optimal way to address learning difficulties by delivering personalized feedback. Usually feedback is not available during modeling activities but is given after a task has been completed. In feedback literature this is referred to as **outcome feedback**, the simplest form of feedback, indicating whether or not results are correct, thus providing minimal external guidance (Butler & Winne, 1995). Several researchers highlighted the effectiveness of more informative types of feedback paired with content-related information that guide the process of cognitive activities (Butler & Winne, 1995; Narciss, 2008; Shute, 2008). Studying the process of conceptual modeling of novice analysts might provide insights on what type of feedback would be effective in **guiding a modeling process** by determining the characteristics of a process of conceptual modeling that have a positive impact on conceptual model quality. Within this study we will therefore focus on revealing the aspects of the modeling process that might affect the quality of a model, and subsequently will formulate our research questions as: “1) *Is it possible to identify patterns of a modeling process that can be associated with better/worse learning outcomes ?* 2) *What type of data is relevant to support the identification of such patterns ?*”.

In order to answer the research questions, we opted for an empirical/experimental approach. Data on modeling activities of novice modelers (86 students in total) have been collected by means of experimental logging functionality of the JMermaid¹ modeling environment. Students' group works over one semester period of time were observed. For data analysis we opted for process mining techniques motivated by the fact that process mining techniques process mining has built a reputation of being capable of analyzing rich data trails and activity streams in various contexts (De Weerd, Schupp, Vanderloock, & Baesens, 2013). In addition, process mining diagrams make it easier to visually extract useful information and quantify relevant properties on process-oriented modeling approaches. We further elaborate the findings with quantitative and qualitative analysis.

While findings showed that certain behavioral patterns can indeed be associated with better/worse outcomes in terms of reaching a satisfactory model quality, further examinations are needed to identify more generic patterns. The results of the study can be used to provide recommendations on **process-oriented feedback**. This study presents first insights to support research on learning analytics (e.g. type of data needed) as well as artificial intelligence (e.g. automation of feedback) in the domain of conceptual modeling. In addition, this study can be inspirational for the application of process-oriented learning analytics outside of the topic of conceptual modeling, as learning event data is becoming more readily available through digital learning systems and other educational information systems.

The remainder of the paper is structured as follows. The second section describes the educational context and assumptions used within this paper. Section 3 gives an overview of related work and the research contribution. Section 4 describes the research method followed by section 5 that describes the data analysis and subsequently reports on the results. Section 6 discusses the contributions and limitations of the work. Finally, section 7 concludes the work proposing some future research directions.

2. Educational context and assumptions

To facilitate further reading of this paper, information on the educational context as well as some basic concepts used throughout the paper will be briefly discussed.

A *conceptual model* (also known as domain model) is a complete and holistic view of a system based on conceptual but precise qualitative assumptions about its concepts "entities" and their interrelationships (Embley & Thalheim, 2012). A conceptual model of an information system is defined as an "abstract model" of an enterprise and conceptual modeling in information systems development as the creation of an enterprise model for the purpose of designing the information system (Wand, Monarchi, Parsons, & Woo, 1995). A model is often represented visually as a diagram, by the use of a modeling language. In this paper the modeling language used is UML (Unified Modeling Language) motivated by the fact that UML is the widely accepted standard used for modeling systems throughout software engineering processes. A UML class diagram is the main structural diagramming approach widely used to visually represent an information system's components and relationships (Szlenk, 2006) that are used both in high level conceptual modeling as well as in more detail for lower level design and documentation of programming code (Berardi, Calvanese, & De Giacomo, 2005; Marshall, 2000; Szlenk, 2006). There are several UML diagramming approaches to capture the dynamic view of a system. Within this study we make use of UML statecharts.

The JMermaid tool used in this work is a conceptual modeling environment that has been developed by the Management Informatics research group at the faculty of Business and Economics, University of Leuven. It uses the UML as modeling language, but underneath it relies on the concepts of MERODE², an Enterprise Information Systems engineering methodology developed at KU Leuven (Snoeck, 2014). MERODE uses a limited subset of UML relevant for conceptual modeling that allows removing or hiding details irrelevant for a conceptual modeling view. The framework is based on three kinds of model views: restricted class diagrams called existence dependency graph (EDG), finite state machine (FMS) and an interaction model to combine the structural and behavioral view in a single model, called object event table (OET). To ensure inter and intra model consistency, the tool makes use of built-in intelligence such as automatic checks, as well as a "consistency by construction" (Snoeck, Michiels, & Dedene, 2003) approach that completes missing model elements automatically. This makes the approach easy to use in an educational context. The tool has been subsequently expanded with an experimental logging functionality to collect data on modeling activities.

¹ <http://merode.econ.kuleuven.ac.be/mermaid.aspx>

² MERODE is an Object Oriented Enterprise Modeling method. Its name is the abbreviation of Model driven, Existence dependency Relation, Object oriented DEvelopment. Cfr. <http://merode.econ.kuleuven.be>

In order to measure the effects of the modeling process on learning outcomes we need to distinguish between worse/better models. In this work we will refer to the quality dimensions of the Conceptual Modeling Quality Framework (CMQF) (Nelson, Poels, Genero, & Piattini, 2012) which is rooted in the seminal framework of Lindland and Sindre (Lindland, Sindre, & Solvberg, 1994) and presents a unified view of conceptual model quality. Within this framework, teaching conceptual modeling involves different types of modeling quality. The final objective is to achieve the capability of producing physical models with high external quality. External model validity - also called *semantic quality*- refers to the level to which the statements in a model reflect the real world in a valid and complete way (feasible completeness, feasible validity) (Lindland, et al., 1994). Within this work we will thus focus on the modeling activities that can potentially affect the semantic quality of a conceptual model.

Since it would not be possible to actually track a human mind in our experiment, the concept of **modeling effort** was used throughout the paper to refer to the prevailing number of specific modeling activities as approximation of modeler's mental effort.

3. Related work

Prior studies on improving model quality have been focusing on the cognitive perspective of the modeling process, model understanding, modeling language knowledge as well as perceptual properties of graphical notation by novices (Mendling, Reijers, & Cardoso, 2007; Moody, 2009; Petre, 1995; Recker, Safrudin, & Rosemann, 2010) as major sources affecting the quality of a modeling process output. However, to our knowledge not many research publications have been written observing the characteristics of a modeling process itself that can potentially affect the modeling process outcome, nor have any empirically validated results been reported on the effects of modeling activities in the context of learning outcomes. Process mining techniques have been applied in a variety of contexts. In the context of modeling approaches previous research was limited to observations on business process modeling (Claes, et al., 2013; Hoppenbrouwers, 2005; Pinggera, et al., 2012), i.e. the process-oriented dimension within conceptual modeling. This study targets at observing a process of a conceptual modeling that combines both data and behavioral dimensions. In particular, we focus on the formalization phase of business requirements in which a novice modeler is faced with a task of constructing a semantically correct conceptual model that reflects the structural and dynamic view of a given domain description. Our approach differs by 1. using logging functionality of a modeling and simulation tools to collect modeling activities by means of recording of a user's interaction within a modeling environment, 2. observation of modeling activities that combine modeling of structural and dynamic views in a single model, 3. observation of a period of one semester of students' behavior by means of their group projects, rather than limiting to one experimental cycle.

Process Mining (van der Aalst, et al., 2009) is a field of research situated at the intersection of the fields of data mining and business process management. Over the last decade, Process Mining has attracted a vast amount of researchers who developed tools, techniques, and methodologies to analyze business processes, thereby not relying on, but rather going well beyond, the application of traditional statistical or data mining techniques, by scrutinizing the underlying execution data captured by information systems. The application of process mining to learner's behavioral data can become valuable assets for different education stakeholders when applied to learning processes and thus delivering tangible insights and decision making input for improving learning, interactions, and outcomes. In addition, process mining diagrams make it visually easier to extract and quantify relevant from the process perspective data (Claes, et al., 2013).

As already stated, studies on *learning quality improvements* are closely related with feedback research (Barber, et al., 2011; Butler & Winne, 1995). While feedback is usually given after a modeling task has been completed, referred to as outcome feedback, indicating whether or not results are correct (Butler & Winne, 1995), in the feedback research the effectiveness of more informative types of feedback that guide the process of cognitive activities is highlighted (Butler & Winne, 1995; Narciss, 2008; Shute, 2008). In this research we aim to improve teaching practices in the area of conceptual modeling by investigating the perspectives of *process-oriented* (rather than outcome-oriented) *feedback*. In particular, within this work we focus on the observation of *modeling patterns (repetition / sequence / alternation / frequency / absence / duration)* that can be associated with better learning outcomes (capability of a student to reach a satisfactory model quality). Since our approach relies on process-related data captured during modeling, this study is also to be situated in the context of learning analytics (R. S. Baker, 2010; R.S. Baker & Yacef, 2009; Romero & Ventura, 2010; Siemens & Baker, 2012; Siemens & Long, 2011), a new research area recognizing the importance of analysis of learner activities for the purpose of understanding and optimizing learning process and outcomes. As this domain is currently in full expansion with increased uptake of analytics tools within higher education institutions (Ali, Hatala, Gašević, & Jovanović, 2012; Fritz, 2011; Santos, 2012), we believe that analysis of behavioral learner data with process

mining can add value in addition to the currently available learning analytics tools and techniques. Ultimately, the results can be further expanded to provide process-oriented guidelines with a focus on tool support for automated feedback which is in the domain of artificial intelligence in education³.

4. Methodology

This work targets a knowledge problem, in particular, our lack of knowledge about how the process of modeling, i.e. modeling activities, can be associated with better/worse learning outcomes. In order to answer our research question we opted for an empirical study approach. Modeling activities of students over one semester period of time have been observed.

4.1. Data collection and sample clustering

During the semester students were assigned a group project, with an approximately 5 page specification document based on real-world requirements (see *Appendix 1*), on which they were supposed to work during the whole semester. They had to analyze and transform the requirement statements into a conceptual model. The project had two deadlines: a deadline for submitting an intermediate solution to receive peer feedback, and the final submission deadline by the end of the semester. Students were randomly assigned into groups with 2-4 students in each. This resulted in 20 observable cases for observing common characteristics of a modeling process. Based on the final score (min. score = 0; max. score = 20) we further classified the cases into best performing and worst performing groups. This resulted in 5 cases in these clusters which we will further refer to as best performing and worst performing groups.

4.2. Capturing events of the modeling process

In order to observe the modeling process (how the novices created their models) interactions with the modeling tool have been logged. As modeling manifests in the creation of modeling elements, in our logs we capture a modeling process as a sequence of create, edit, delete, undo, redo, copy events. These events are further abstracted into CREATE and EDIT (grouping events edit, delete, undo, redo, copy) representations.

Table 1: Sample format of logs, each row representing one event. The complete data set contains 10.000 events from 20 groups)

TIMESTAMP	GROUP ID	SESSION ID	SESSION TYPE	SCORE	ORIGINAL ACTIVITY	ABSTRACTED ACTIVITY	MODELING VIEW	DIAGRAMMING TYPE
19/11/2013 1:54:00	1	Session1	EARLY	6	CREATE OBJECT	CREATE	S	EDG
19/11/2013 1:54:16	1	Session1	EARLY	6	CREATE OBJECT	CREATE	S	EDG
19/11/2013 1:55:55	1	Session1	EARLY	6	CREATE DEPENDENCY	CREATE	S	EDG
19/11/2013 2:08:03	1	Session1	EARLY	6	CREATE ATTRIBUTE	CREATE	S	EDG
19/11/2013 2:08:36	1	Session1	EARLY	6	CREATE EVENT	CREATE	B	OET
19/11/2013 4:37:28	1	Session1	EARLY	6	CREATE EVENT	CREATE	B	OET
19/11/2013 4:40:05	1	Session1	EARLY	6	DELETE EVENT	EDIT	B	OET
19/11/2013 4:40:18	1	Session1	EARLY	6	UNDO DELETE EVENT	EDIT	B	OET
19/11/2013 5:09:53	1	Session1	EARLY	6	REDO DELETE EVENT	EDIT	B	OET
19/11/2013 5:10:58	1	Session1	EARLY	6	EDIT ATTRIBUTE	EDIT	S	EDG
10/12/2013 11:04:18	1	Session2	LATE	6	CREATE METHOD	CREATE	B	OET
10/12/2013 11:04:23	1	Session2	LATE	6	CREATE STATE	CREATE	B	FSM
10/12/2013 11:05:05	1	Session2	LATE	6	CREATE STATE	CREATE	B	FSM
10/12/2013 11:05:23	1	Session2	LATE	6	CREATE TRANSITION	CREATE	B	FSM
12/12/2013 11:02:34	1	Session3	LATE	6	DELETE DEPENDENCY	EDIT	S	EDG
12/12/2013 11:02:41	1	Session3	LATE	6	CREATE DEPENDENCY	CREATE	S	EDG
12/12/2013 11:02:44	1	Session3	LATE	6	CREATE ATTRIBUTE	CREATE	S	EDG
12/12/2013 11:02:47	1	Session3	LATE	6	DELETE STATE	EDIT	B	FSM
12/12/2013 11:03:00	1	Session3	LATE	6	EDIT TRANSITION	EDIT	B	FSM

Events will be defined at different levels of abstraction by making use of one or more attributes (columns in Table 1). This will enable the analysis of different aspects of the modeling process in addition to easier to understand visualizations. For example, at a high abstraction level, we can specify events according to the conceptual modeling view, i.e. whether they can be attributed to modeling structural (S) or behavioral (B) characteristics. For a more specific view, event names can be supplemented with other information fields, for instance with the modeling diagram type where the class

³ <http://www.nmc.org/pdf/2010-Horizon-Report.pdf>

diagram⁴ (S//EDG) represents the creation of business objects and their properties, the object event table (B//OET) represents business events and rules for interactions, and finite state machines (B//FSM) represent lifecycles of objects with sequence constraints as basis for dynamicity of a system. At the most detailed level, we can also make use of the original activity name (potentially in combination with other information), for a fine-grained analysis of modeling activities and patterns. Groups are identified by group id and the grade they obtained for the final solution (column 'score' in Table 1).

Finally, so as to be able to distinguish between modeling phases, we further identified sessions based on the event's timestamp to make distinction between EARLY and LATE sessions. This allows identifying data at two different levels: based on (1) group id and (2) session level, by combining group id and session id. An extract of the data set is presented in Table 1.

4.3. Three-dimensional analysis

Event logs of students' group works have been analyzed using process mining techniques. Event data of 20 cases (10.000 events in total) have been subjected to a three-dimensional analysis (see further).

- 1) **Hierarchical**: an investigation of top-level models discovered from the data where cases are regarded as sequences of structural (S) and behavioral (B) activities (either fully abstracted or appended with CREATE/EDIT), followed by a session-level, fine-grained analysis of both structural and behavioral activities in isolation.
- 2) **Modeling performance**: a contrast analysis was performed to identify differences between best and worst scoring groups.
- 3) **Time trend analysis**: by making a distinction between "early" and "late" sessions.

This analysis was carried out using Excel, Disco, and ProM. From a process mining perspective, we could make use of two prominent techniques: (1) process model discovery using Disco⁵ and (2) Dotted chart analysis in ProM (van der Aalst, et al., 2009). We further elaborate the findings with quantitative and qualitative analysis.

5. Data Analysis

5.1. Subjects and sample representativeness

The study was conducted in the context of the course "Architecture and Modeling of Management Information Systems"⁶ with participation of 86 students randomly assigned to 20 groups. The course targets at master level students with heterogeneous backgrounds from the Management Information Systems program at the KU Leuven. The goal of the course is to familiarize the students with modern methods and techniques of Object-Oriented Analysis and Design for Enterprise Information Systems and to let them acquire sufficient skills of developing an enterprise model as basis of an enterprise information system. Analysis of the personal characteristics of the students resulted in the demographics presented in Table 2.

Table 2: Summary of demographics

Gender	
Male	74 %
Female	26 %
Age distributions	
Min age	21 year
Max age	42 year
AVG age	25,6 year
<= 25	62 %
25 < 35	28 %
>= 35	10 %
Previous knowledge of data modeling	
No knowledge	30 %

⁴ In MERODE we refer to class diagram as existence dependency graph (EDG) due to the fact that relationships are translated into existence dependencies

⁵ Disco is a commercial tool developed by Fluxicon: <http://fluxicon.com/disco/>

⁶ The course's page can be found on <http://onderwijsaanbod.kuleuven.be/syllabi/e/D0I71AE.htm>

<i>Little knowledge</i>	33 %
<i>Moderate knowledge</i>	29 %
<i>Extensive knowledge</i>	8 %
Scores of group works (on 20 scale)	
<i>Min score</i>	5
<i>Max score</i>	19
Total participants	86

5.2. A bird's eye view on the modeling process

The first step in our analysis approach consists of the creation of a top-level model. We filtered the dataset based on group number as ID, timestamp, activities (Create New Model, modeling activities targeting the structural view (S), modeling activities targeting the behavioral view (B)). Although the high level of abstraction prevents the extraction of ample insights, the fully abstracted view on the modeling process of novice business analysts does reveal a number of interesting patterns from a quantitative perspective: in a majority of cases, modeling of structural aspects was found to precede the activities for modeling the behavioral aspects of the system (Figure 1, Figure 2, Figure 3). We could partially attribute this to the sequence of topic distribution within the course: students first learned techniques for analysis and modeling of the structural aspects of the system followed by modeling dynamic aspects of a system in later sessions of the course. Activities for modeling the behavioral aspects exceed the number of activities for modeling the structural aspects (approximately 60% of all activities) (Figure 1). This pattern holds true in both best performing and worst performing clusters (Figure 2, Figure 3). The presence of cycles between modeling activities for structural and behavioral views in all three graphs indicates the non-linear (i.e. iterative) modeling process pattern over the entire period of observation. Between group analysis reveals almost twice as many occurrences of modeling activities (both for modeling structural and behavioral aspect) in the best scoring cluster as compared to the worst scoring cluster.

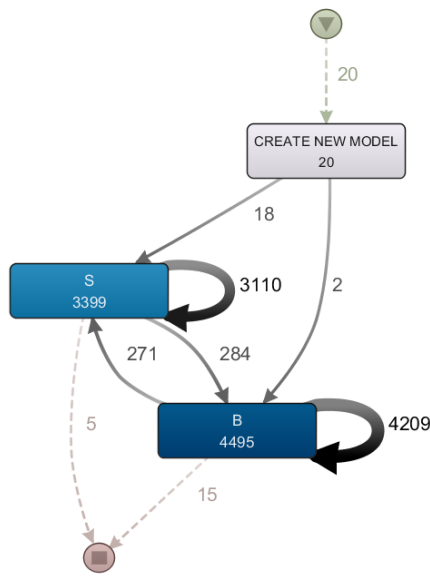


Figure 1. Top level model - All 20 cases

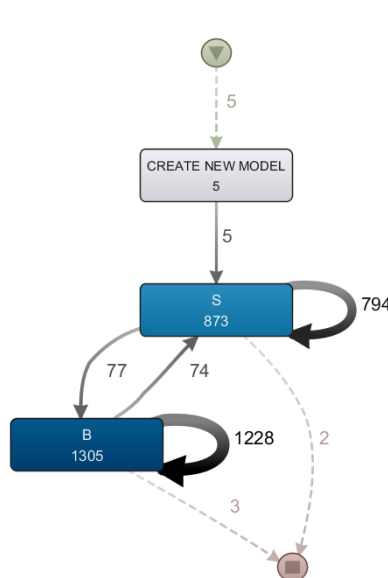


Figure 2. Top level model - Best 5 cases

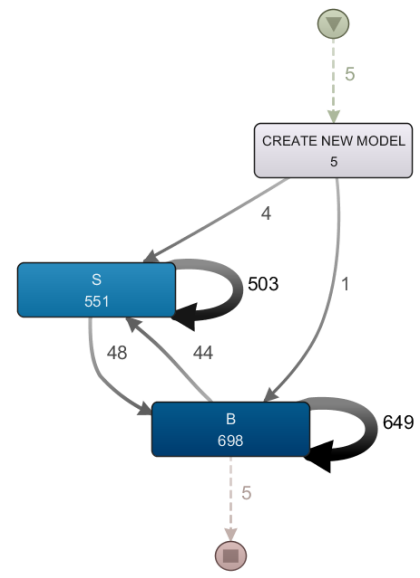


Figure 3. Top level model - Worst 5 cases

In summary, the top level process diagrams suggest the following conclusions: 1. The modeling of business requirements has an iterative character over time; 2. The modeling of the structural view is a leading activity when transforming business requirements into formal models; 3. In general, the number of activities for modeling of behavioral aspects prevails over the number of activities for modeling the structural view of a system, the proportion in both best and worst scoring groups is the same (60 % of design activities); 4. The analysis of the graphs shows a higher (almost double) frequency of modeling activities for the same time period in the best scoring group. This suggests that the more students were engaged in modeling activities the better the modeling process output became.

5.3. Session level analysis

Next, zoomed into the modeling process by considering the activities that happen in the context of a single session, i.e. what happens between opening and closing a projects file. We therefore added event types based on a CREATE/EDIT typification which resulted in the process diagrams shown in Figure 4, Figure 5, Figure 6.

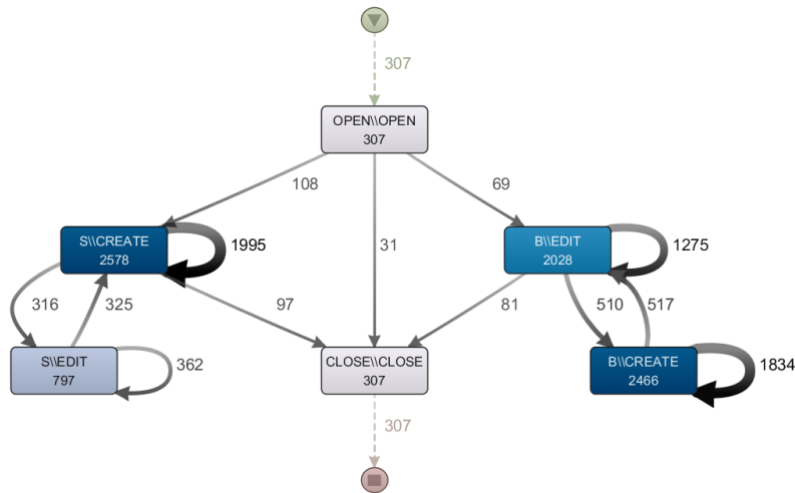


Figure 4. Top level model - All 20 cases (create & edit activities)

The diagrams show that the linear approach of modeling is prevailing within each session both in best and worst performing clusters, i.e. students preferred to concentrate on one task at a time working on either structural or behavioral aspects of a system within one open/close-delimited session. No modeling activities were found in 31 out of 307 sessions. This might indicate that students used this sessions to just view their model solutions presumably for verification and validation purposes. We will further refer to sessions that do not contain modeling activities as “view” sessions.

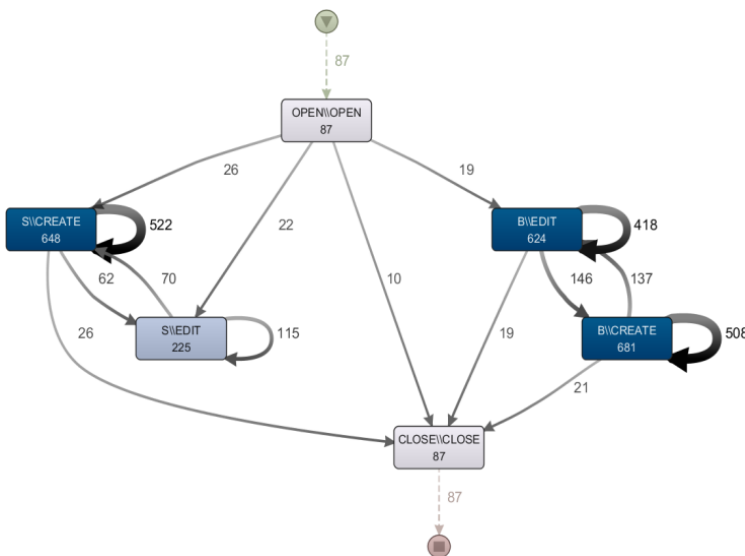


Figure 5. Top level model – Best 5 cases (create & edit activities)

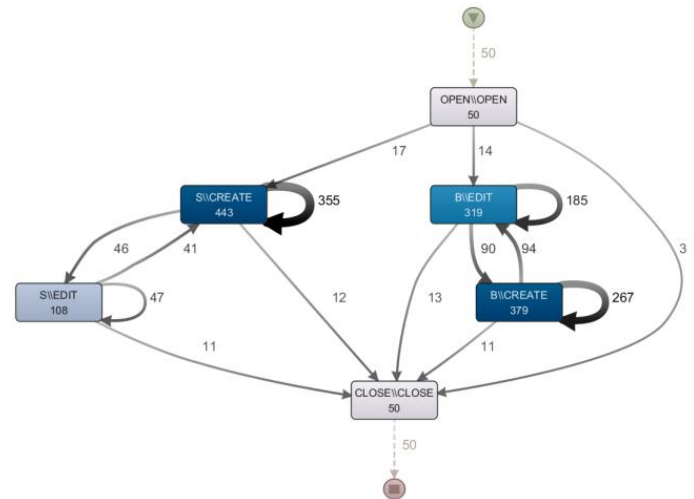


Figure 6. Top level model – Worst 5 cases (create & edit activities)

The most prevailing difference between both groups is the “effort” put into the behavioral view: almost double the amount of structural events. In contrast to the worst performing groups the best performing groups seem to have more “view” sessions (10 out of 87 versus only 3 out of 50), presumably for validation and verification purposes.

5.4. Between-session analysis based on diagram types

Next, we further subdivided the structural and behavioral views by adding event types based on the S/EDG, B/OET, B/FSM typification which resulted in the process diagrams shown in Figure 7 and Figure 8. As found from within-session analysis, students worked on structural and behavioral views in sequence. To neutralize the effect from topic distributions in the teaching process that might presumably affect a modeling approach, we filtered away the open/close events to observe the patterns over the entire process of the semester. In contrast to within-session diagrams such between-session analysis allowed to reveal some distinctions between best and worst performing groups. The process diagrams that capture the modeling behavior of the of the entire semester (with open/close events filtered away) show that (1) the linear modeling pattern still holds true within the worst performing groups for the entire semester, i.e. they are inclined towards modeling “one task at a time”, with structural modeling actions taking place before behavioral modeling with little revisiting activities from behavioral modeling to structural modeling (see Figure 7) thus revealing a more *linear* character of modeling approach in the worst performing groups (2) in contrast, the process diagram for the best performing group shows alternations between modeling activities, i.e. students worked on modeling structural and behavioral views in parallel or revisited and adapted different views of the system afterwards (see Figure 8). This seems to suggest that they were more verification and cross-validation oriented, thus conveying a more *iterative* modeling approach.

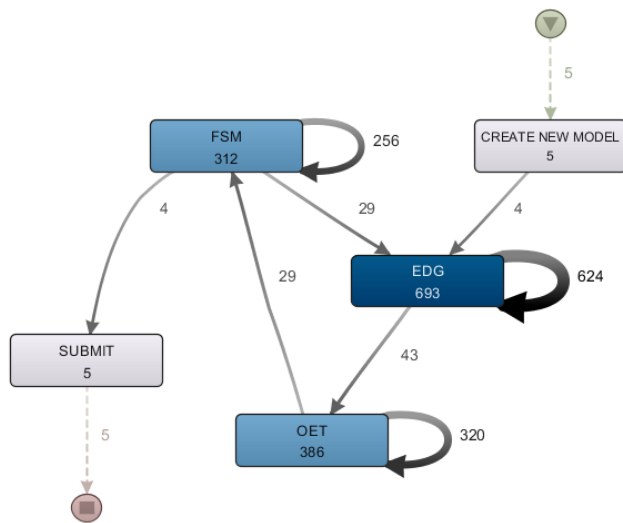


Figure 7. Worst 5 cases (class diagram, business events & sequence constraints)

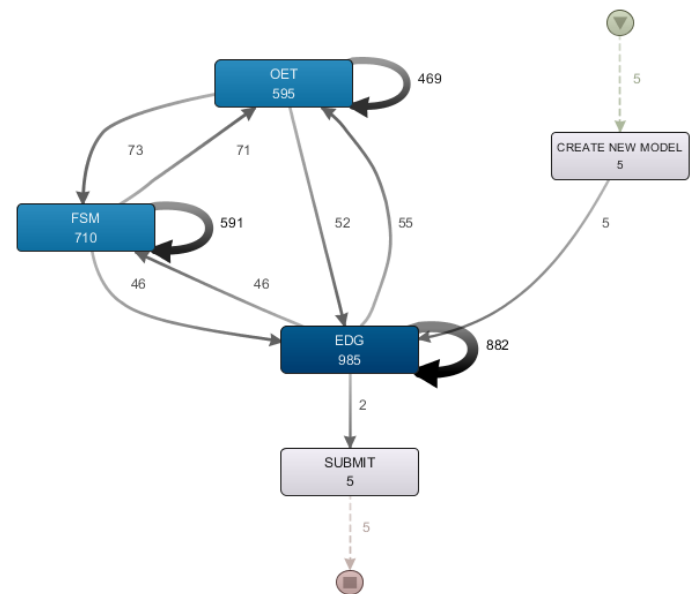


Figure 8. Best 5 cases (class diagram, business events & sequence constraints)

5.5. Time trend analysis of early vs. late sessions

Next, we performed a session level analysis but additionally subdivided the modeling period into early and late sessions which resulted in the process diagrams shown in Figure 9, Figure 10, Figure 11, Figure 12. Within worst performing groups designing and editing activities still shows more isolation between structural and behavioral modeling (Figure 10), whereas in the best performing groups editing of either structural or behavioral aspects conforms to the aforementioned tendency of switching modeling activities between the structural and behavioral views (Figure 9). The diagrams also show that the best performing groups were more active in earlier sessions (largest part of their create activities) while the number of modeling activities showed a tendency to decrease in later sessions (Figure 11). It seems the students first targeted at capturing relevant information from textual requirements into their model (prevailing number of create events) and continued to adapt the model in later sessions (prevailing number of edit activities). In contrast to the worst performing groups the best performing groups seem to have more “view” sessions in later sessions. In contrast to the best performing groups worst performing groups remained active in later sessions, with a larger number of create events for the behavioral part compared to their early sessions (Figure 12). In particular, the diagrams suggest that they seemed to have more difficulties with modeling behavioral aspects. In contrast, the best

performing groups seem to use the later sessions to revise either the structural or behavioral views, but to not need to iterate between the two dimensions any more.

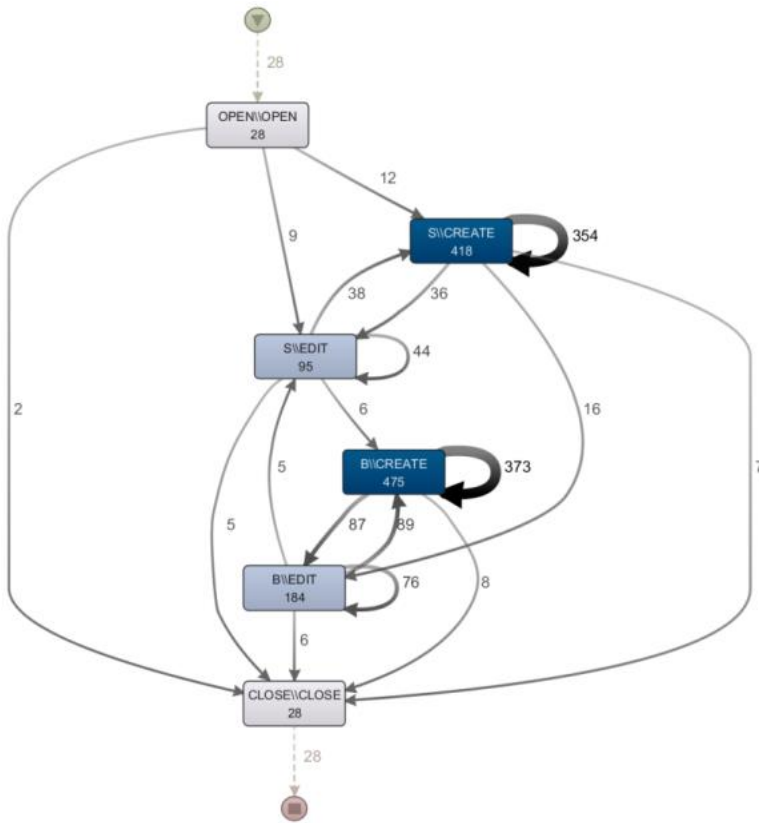


Figure 9. Early sessions modeling Activities (create & edit) – Best 5 cases

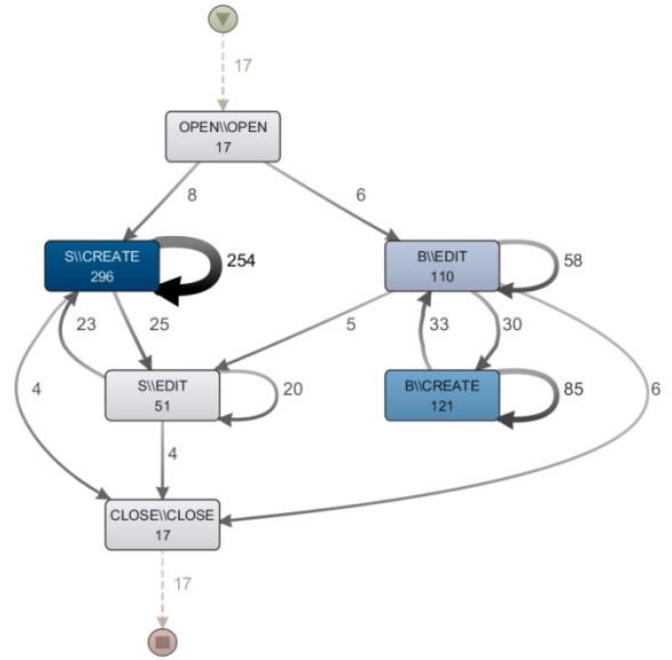


Figure 10. Early sessions modeling Activities (create & edit) – Worst 5 cases

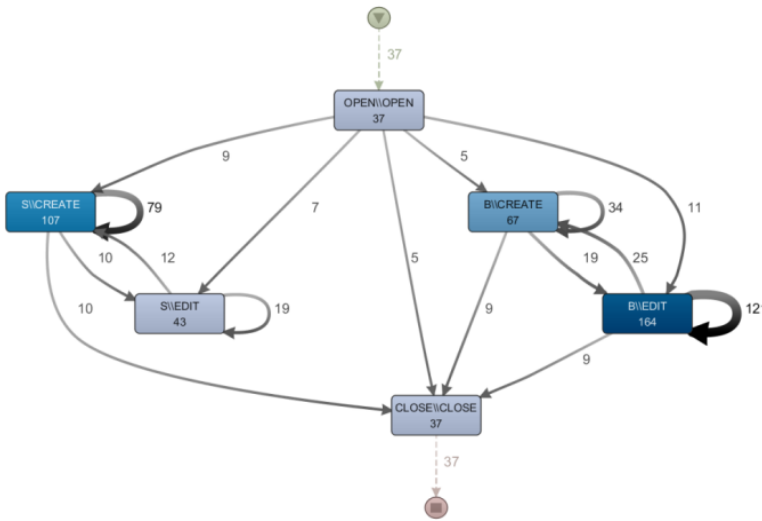


Figure 11. Late sessions modeling Activities (create & edit) – Best 5 cases

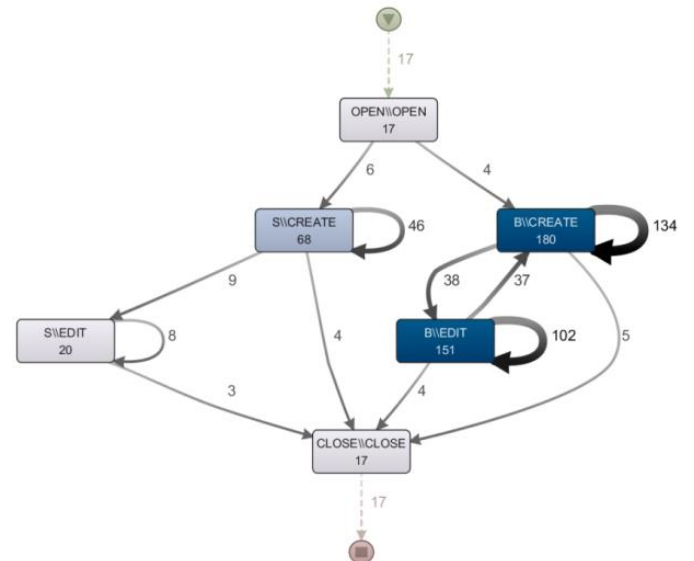


Figure 12. Late sessions modeling Activities (create & edit) – Worst 5 cases

5.6. Time trend analysis of early vs. late sessions with element type information

To better understand the differences between early and late sessions, we “zoomed” into the sessions by detailing create-events in each view with the type of element being created, i.e. object, attribute, dependency, inheritance, dependency, event, state chart, state, transition, or constraint. This resulted in the process diagrams presented in Figure

14, Figure 15, Figure 16, Figure 17 (see Appendix 2). The diagrams show that the best performing groups seem to capture more information through the use of objects: 149 occurrence vs. 92 in worst performing groups. The best performing groups also differ in terms of lesser usage of attributes: 40 vs 69 in worst groups. This corresponds to the phenomenon that students in the worst performing group often had too many attributes that duplicate information that was already present in the model under the form of associations or states. E.g. they would add an attribute “student number” in a class REGISTRATION, while this class had a 1-1 link to the class STUDENT. The best performing groups seem to stand out by using more advanced modeling concepts. For example, inheritance was used 13 times in these groups versus only 2 times in worst performing groups. Students in the worst performing group seem to be reluctant to use such more advanced concept because they don’t master it well while having been told that it is difficult to use well. Best performing groups seem to detail more on behavior in early sessions: creation of a business event and methods (62 and 71 vs 24 and 38 in worst performing groups), creation of state machines (34 vs 11 in worst performing groups), creation of states (121 vs 17 in worst performing groups), creation of transitions (187 vs 31 in worst performing groups). In the late stages of modeling, activities in the best performing groups seem to decrease both for structural and behavioral aspects while the quantitative analysis in worst performing groups shows a continuously active modeling process. This seems to indicate that later phases in best performing groups had already reached a satisfying solution and presumably had a more verification/validation-oriented character, as can be deduced from their more extensive use of constraints as well as the availability of pure “view” sessions (they only “looked” at the model, presumably during their verification/validation activities). The proportions in better and worst performing groups seem to indicate that best performing groups didn’t make substantial changes to the structural view but rather detailed it with more use of attributes. This explains the absence of switching between structural and behavioral modeling noticed in the previous paragraph: detailing a model with attributes will not require revising the behavioral model for consistency. Modeling of behavioral aspects prevailed over the structural view in best performing groups but in terms of quantity these activities were still significantly less frequent than in worst performing groups: creation of a business event and methods (14 and 21 vs 35 and 50 in worst performing groups), creation of state machines (3 vs 11 in worst performing groups), creation of states (4 vs 18 in worst performing groups), creation of transitions (25 vs 66 in worst performing groups). This seems to indicate that best performing groups understand how the existence dependency graph already captures a lot of behavioral aspects of the domain, hence requiring only little additional FSMs. The verification/validation tendency in the latest phases of modeling in best performing groups was confirmed by the use of constraints embedded in structural view (5:0).

5.7. Distributions of modeling effort over time

We next applied a dotted chart diagram to observe the differences in terms of frequencies and gaps between session activities over the semester (Figure 13). The context information necessary to read the figure is presented in Table 3. The dots show the modeling sessions in two colors: green (prevailing number of structural aspects) and blue (prevailing number of behavioral aspects). The red dots represent the “create new model” event (the first event in every case) which in some cases are shown because there were no modeling activities in the first session, but not visible in other cases because of overlap. The worst performing groups are indicated by red arrows, and the best performing groups by green arrows. Context information is shown by vertical arrows labeled by abbreviations from Table 3.

Table 3: Context information

<i>Assignment available (AA)</i>	08/11/2013
<i>Deadline for intermediate solution (D1)</i>	22/11/2013
<i>Exercise session on testing peer solution (T)</i>	22/11/2013
<i>Peer feedback deadline (PD)</i>	02/12/2013
<i>Last class (LC)</i>	16/12/2013
<i>Deadline for solution (D2)</i>	20/12/2013

The number of modeling sessions in the worst performing groups were limited to 2-3 with significant gaps between the sessions. In contrast, best performing groups were distinguished by more frequent modeling sessions. The context information (see Table 3) shows that both best and worst performing groups were sensitive to deadlines (submission for peer review and final submission) in terms of being more intensively engaged in modeling activities right before the deadlines. In the best performing group however more intensive activities were also found in between the deadlines. In addition after the peer review best scoring groups seem to react to the given feedback and adapt their models to

address the comments, while the worst performing groups didn't "react" to peer comments. This also confirms that worst performing groups were mostly ignorant about model validity.

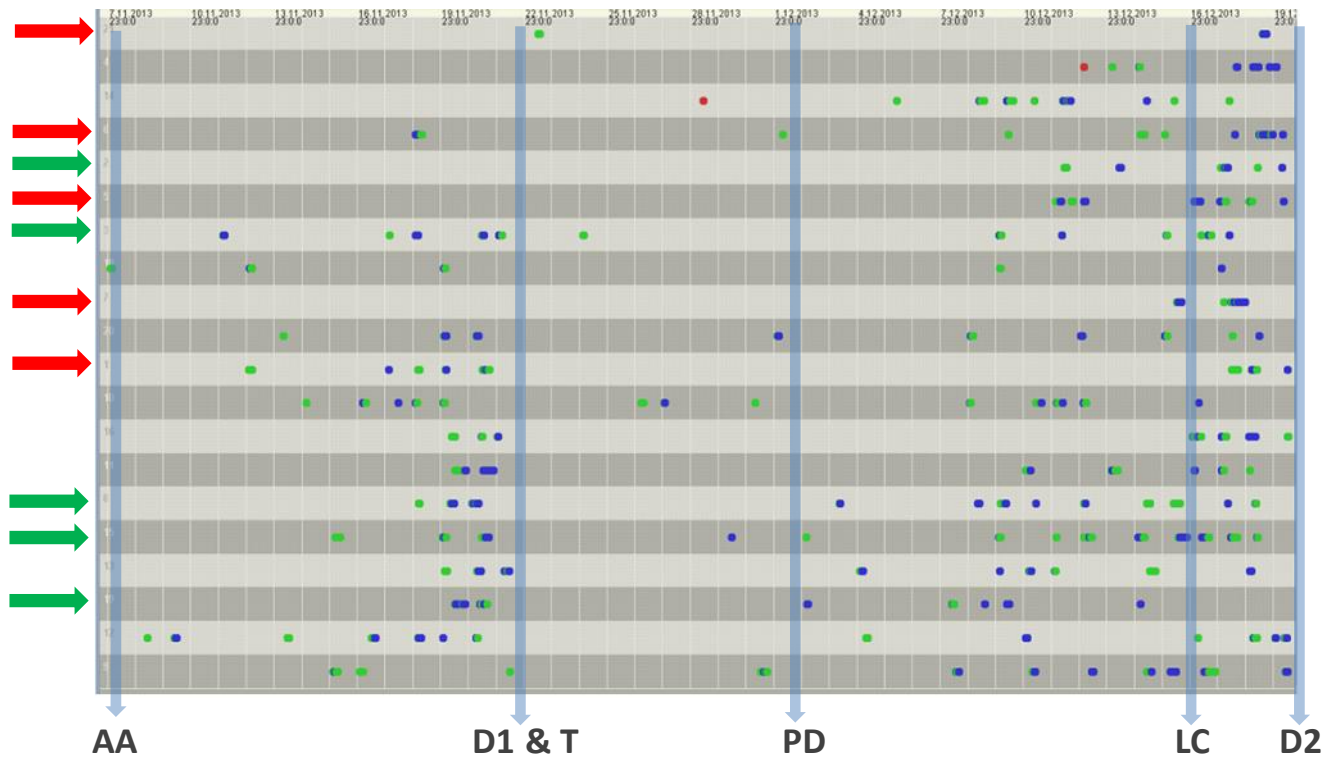


Figure 13. Distribution of modeling effort over time

5.8. Other observations

While examining the model solutions in a qualitative manner, we found that best scoring groups were oriented towards more extended class diagrams to capture information with a more extensive use of classes. In contrast, worst performing groups have smaller class diagrams (with less classes) but have a more extended use of attributes and states to capture the required information. In addition worst performing groups seem to be unaware of the trade-off that needs to be made between information needs over time versus short term solutions. As an example a best performing group would keep a class ROOM to capture "availability of a room" over time. This allows e.g. keeping historical data on room reservations and checking the availability of a room before it can be reserved. On the other hand, worst performing groups would limit themselves to an attribute room in a class EXAM, which doesn't allow tracking historical information or room availability through time. Worst scoring groups were found to make heavy use of attributes when capturing information (169 attributes for 69 objects, vs. 171 attributes per 90 objects). Worst performing groups in addition were distinguished by the use of redundant use of constraints, attributes or states, e.g. they supply attributes "isModifiable" and "isNotModifiable" in the same class, while either one would be enough. Worst performing groups also seem to have difficulties in distinguishing between the need to capture information via an attribute vs. a state of an object. They often would provide an attribute (e.g. attribute "isCanceled") instead of detailing the behavior by means of defining a state (e.g. state "canceled" and a business event "cancel" to allow a transition to that state). Often even both an attribute and state were defined. Very often they seem to add modeling constructs in order to solve experience problems in their solution, rather than conceptually thinking about 'what is required'. One could compare this to a plumber adding lots of pipes and joints to an existing system, rather than rethinking the entire design of the plumbing. This kind of plumbing approach reveals itself in adding lots of states and transitions that do not reflect the real business domain, hence resulting in a wrong use of the concept of 'business event'. This approach is also confirmed by students' approaches focusing first on secondary properties in early sessions of modeling (e.g. extensive use of attributes).

6. Discussion: contributions, findings and limitations

In the domain of conceptual modeling, not many empirical studies can be found that investigate if and how the process of modeling can be associated with learning outcomes. Previous studies focused on cognitive perspective of modeling

process, model understanding, modeling language knowledge as well as perceptual properties of graphical notation by novices as major sources affecting the quality of a modeling process output. Yet those studies did not investigate the effects of modeling process on the modeling outcome. The largest number of experimental studies are to be found in the domain of business process modeling. Experimental studies in this domain however were limited to one cycle experiment using paper exercises. This study addresses this gap of lack of empirical studies by using a tool support to log learner's activity over longer period of time.

From a theoretical perspective this work presents three major contributions: (1) the results contribute to improving our knowledge on the process-related aspects of conceptual modeling that can be associated with the quality of a modeling process outcome, namely a model quality; (2) the results provide empirical support for the use of process oriented feedback in the domain of conceptual modeling; (3) The paper also suggests a novel approach for analyzing behavioral learner data through the application of process mining techniques which opens up new perspectives for learning analytics and artificial intelligence research in the domain of conceptual modeling.

The results of the work show that certain patterns in the modeling process of novices can indeed be indicative for the quality of a conceptual model. The most frequent scenarios were synthesized into process patterns presented below. Based on the findings we provide recommendations for (1) teaching guidance to improve process-oriented feedback, (2) logging needs to support further research on learning analytics in the field of conceptual modeling.

6.1. General findings: patterns

- a. Conceptual modeling processes were found to be iterative over time with alternations between modeling the structural and behavioral view of the system.
- b. More modeling effort (number of modeling activities) presumably leads to better modeling process outcomes.
- c. Modeling the behavioral aspects of a system seemed to require more effort than modeling the structural view: it requires almost double the amount of effort put into structural view.
- d. Within sessions analysis showed that adapting the behavioral view of the system required more effort than adapting a structural view of a system (60% of modeling activities with prevailing edit activities).
- e. Novices' iterations were limited within sessions (focusing on either structural or behavioral modeling activities in one session). However, over time the modeling behavior of worst performing groups was characterized by a linear approach (one modeling task at a time), while the modeling behavior of best performing groups was characterized by an iterative pattern (frequent switches between different modeling views).
- f. Session-based analysis showed that best performing groups first targeted at capturing the most relevant information from textual requirements into their model (prevailing number of create events) and continued to adapt the model in later sessions (prevailing number of edit activities). Worst performing groups in contrast seemed to focus more on secondary properties in early sessions of modeling (extensive use of attributes) and continued to actively expand the model in later sessions (higher number of create events).
- g. Modeling the behavior of a system in best performing groups had a more isolated and independent character (i.e. focusing on what the system needs to do with a prevailing number of create events in earlier sessions), while the worst performing groups seemed to have more difficulties with modeling behavioral aspects. The additional qualitative analysis revealed the *more reactive* approach of worst performing groups (focusing on correcting model errors) by adding events to support transitions to/from states which are not explicitly required by business requirements statements.
- h. In general modeling activities in best performing groups were characterized by a tendency to decrease over time. In contrast, modeling activities in worst performing groups showed a tendency to increase.
- i. Analysis of effort distribution over time showed that worst performing groups were less active in terms of frequency of modeling sessions with significant gaps between the sessions. In contrast, best performing groups were distinguished by more frequent modeling sessions.
 - i. The context information showed that both best and worst performing groups were sensitive to deadlines (submission for peer review and final submission) in terms of being more intensively engaged in modeling activities right before the deadlines. In the best performing group however more intensive activities were also found in between the deadlines.

- ii. Best scoring groups were found to be more reactive to peer feedback and were eager to adapt their models in accordance to feedback, while the worst performing groups were found to be ignorant about peer feedbacks.

6.2. Recommendations from the teaching perspective: sample process-oriented feedback

From the teaching perspective with regard to a modeling process the findings suggest that students can be advised to first concentrate on identifying the relevant information to be captured in a business model, i.e. business concepts (such as business objects and business events) into flat lists, without relating them as is done through class diagramming and state charts. In that way students can concentrate on understanding the requirements first and avoid completing their lack of domain knowledge by wrong interpretation or imaginary representation of a domain. To fill the gaps in domain knowledge caused by insufficient thorough reading of the requirements document, students tend to revert to their knowledge of similar domains from the real world. For example, when a task is related to a customer service in a bank, a student associates it with a representation of a domain based on own experience and “completes” the requirements with “imaginary requirements” which were not required by the original requirements text. By capturing business concepts in ‘flat lists’ it is less likely that the initial analysis process interferes with the lack of modeling knowledge and lack of modeling language knowledge. Presumably, the misinterpreted use of a modeling construct which is discovered later on causes the “reactive” modeling pattern. Frequent verification and validation activities should be stimulated (e.g. by means of peer feedback, exercises on testing a model, etc.). If longer periods are considered, the use of deadlines seems to have a positive effect in terms of stimulating students’ engagement in modeling activities.

6.3. Recommendations for recording learner data logs

The findings suggest that logging at the tool interaction level will provide more information, rather than at the modeling level. Of course modeling activities are central, but they are inherently part of tool interaction, while this is not the case the other way around. Addressing the logging functionality limitations, such as absence on records on distractions from modeling activities, verification and validation activities (e.g. simulation of a model), thinking and viewing activities will allow more thorough examination of conceptual modeling process. Viewing a model for verification purposes may not necessarily lead to modifications in a model and is therefore absent in the current logs. However, logging viewing activities can be used to investigate whether a student cross-validated the modeling views by simply inspecting different views visually. Other interesting activities to log include the observation of feedback (e.g. model checks, execution/simulation of a model) in order to better observe the knowledge generation process affected by automated feedback during the modeling process. This type of information can, for example, be used to investigate how a modeler reacted to a feedback (e.g. “it was the intended behavior I wanted to check” or “a model needs to be modified to address a detected problem”).

6.4. Limitations

To facilitate further research and in particular for future experiments in this domain, certain limitations of this study should be mentioned. This study is limited to student sample from one particular course from one university. For more patterns it would be interesting to compare the results with those obtained by using student samples from other universities and using diagramming techniques other than UML class diagrams and statecharts. Individual rather than group works can be studied to observe the effects from personal characteristics that might be relevant (e.g. for personalization of feedback) such as gender, previous knowledge of data modeling, the level of computer self-efficacy level in terms of their ability to learn and use a computer software and general ICT experience in terms of previous programming experience (Compeau & Higgins, 1995; Davis, 1989; Davis, Bagozzi, & Warshaw, 1989; Keller, 2009; Poelmans & Wessa, 2013; Venkatesh, Morris, Davis, & Davis, 2003). Finally, comparison between novices and experts rather than best/worst solutions could provide more insights on modeling patterns.

To achieve even better insights traceability with requirements would be needed. This would however imply the use of natural language processing (machine learning) techniques to trace the semantic quality of a model based on the textual description. The use of such techniques in the context of conceptual modeling is not yet mature enough to be applied in such kind of experiments. As an example among the challenges of machine learning algorithms are the identification of co-reference (Pradhan, et al., 2011), modality and negation (Chapman, Bridewell, Hanbury, Cooper, & Buchanan, 2011; Wiebe, Wilson, Bruce, Bell, & Martin, 2004), not speaking about the ambiguity and inaccuracy that exist in natural language. As an example, in the requirement statement “Papers need to be reviewed by at least three reviewers;

however they must be registered in a system before being assigned to a reviewer”, it would be challenging to identify to whom/what the word “they” refers to (papers or reviewers) using machine learning techniques.

7. Conclusions and future work

Feedback is central to the research on improving learning achievements. While feedback for modeling activities is usually available to novices when a modeling task is complete, this research aimed at revealing perspectives of process-related feedback in the domain of conceptual modeling by examining if certain characteristics of a modeling process can affect the output of a modeling process- the quality of a model. Modeling activities of novices were empirically examined to find if certain modeling process patterns could be indicative for better/worse learning outcomes based on a differentiation of the semantic quality of a model. In this study we used a novel approach for analyzing modeling activities of novices by means of application of process mining techniques. While findings showed that certain behavioral patterns indeed can be associated with better/worse learning outcomes, further examinations are still needed to identify more generic patterns. The study proved to serve a promising starting platform for process-oriented research and feedback in the field of conceptual model quality. The results also provide first insights for research on learning analytics and artificial intelligence in the domain of conceptual modeling. Ultimately the study can support research on interactive learning environments to stimulate learner motivation and engagement (Jou, Chuang, & Wu, 2010) in the domain of conceptual modeling as well as artificial intelligence (e.g. feedback automation).

As a further extension experiments with a modeling environment with expanded logging functionality can be used. In particular, we plan to observe the effects of feedback incorporated in the modeling environment such as the use of built-in intelligent features students can apply during the process of modeling, e.g. to check intra/inter model consistency and to make use of the combined logs with the logs of feedback-enabled simulation of models (Sedrakyan, Snoeck, & Poelmans, 2014). The latter feature allows student to execute their models in order to validate the semantic quality of a model and provides automated feedback that visually links the test results to their causes in the model's design. This allows detecting design errors that result from misinterpreted use of modeling language constructs. By logging the interactions with this feature we can also observe the effects of process-oriented feedback and knowledge generation process throughout modeling activities in order to optimize and personalize feedback. Another possible direction could be proceeding into other relevant learning analytics research targets in the domain of modeling, such as investigating (1) actions that can indicate engagement, motivation and satisfaction, (2) features of the modeling environment that may lead to better outcomes, (3) if/when students are ready to move to the next topic, (4) if a student is at risk of not completing the course successfully and/or should receive help, (5) what grade a student is likely to receive without intervention, etc. Comparisons with expert modelers' modeling process patterns could provide even better insights.

References

- Ali, L., Hatala, M., Gašević, D., & Jovanović, J. (2012). A qualitative evaluation of evolution of a learning analytics tool. *Computers & Education*, 58(1), 470-489.
- Baker, R. S. (2010). *Data Mining for Education* (3 ed. Vol. 7): Oxford, UK: Elsevier.
- Baker, R. S., & Yacef, K. (2009). The state of educational data mining. *A review and future visions". JEDM-Journal of Educational Data Mining*, 1(1).
- Barber, L. K., Bagsby, P. G., Grawitch, M. J., & Buerck, J. P. (2011). Facilitating Self-Regulated Learning With Technology: Evidence for Student Motivation and Exam Improvement. *Teaching of Psychology*, 38(4), 303-308.
- Berardi, D., Calvanese, D., & De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1), 70-118.
- Butler, D. L., & Winne, P. H. (1995). Feedback and Self-Regulated Learning: A Theoretical Synthesis. *Review of Educational Research*, 65(3), 245-281.
- Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., & Buchanan, B. G. (2011). A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5), 301-310.
- Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H. A., Weber, B., & Poels, G. (2013). A visual analysis of the process of process modeling. *Information Systems and e-Business Management*(March), 1-44.
- Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2).

- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340.
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8), 982-1003.
- De Weerd, J., Schupp, A., Vanderloock, A., & Baesens, B. (2013). Process mining for the multi-faceted analysis of business processes - a case study in a financial services organization. *Computers in Industry*, 64(1), 57-67.
- Embley, D. W., & Thalheim, B. (2012). *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*: Springer.
- Endres, A., and H. Dieter Rombach. (2003). A handbook of software and systems engineering: empirical observations, laws and theories. In: Pearson Education.
- Fritz, J. (2011). Classroom walls that talk: Using online course activity data of successful students to raise self-awareness of underperforming peers. *The Internet and Higher Education*, 14(2), 89-97.
- Hadjerrouit, S. (2005). Object-Oriented Software Development Education: a Constructivist Framework. *Informatics in Education*, 4(2), 167-192.
- Hoppenbrouwers, S. J. B. A., HA Erik Proper, and Th P. van der Weide. (2005). A fundamental view on the process of conceptual modeling. In *Conceptual Modeling—ER 2005* (pp. 128-143): Springer Berlin Heidelberg.
- Jou, M., Chuang, C.-P., & Wu, Y.-S. (2010). Creating Interactive Web-Based Environments to Scaffold Creative Reasoning and Meaningful Learning: From Physics to Products. *Turkish Online Journal of Educational Technology - TOJET*, 9(4), 49-57.
- Keller, C. (2009). User Acceptance of Virtual Learning Environments: A Case Study from Three Northern European Universities. *Communications of the Association for Information Systems*, 25.
- Lauesen, S., and Otto, Vinter. (2001). Preventing requirement defects: An experiment in process improvement *Requirements Engineering, Springer-Verlag London Limited*, 6(1), 37-50.
- Lindland, O. I., Sindre, G., & Solvberg, A. (1994). Understanding quality in conceptual modeling. *Software, IEEE*, 11(2), 42-49.
- Marshall, C. (2000). *Enterprise modeling with UML: designing successful software through business analysis*: Addison-Wesley Professional.
- Mendling, J., Reijers, H. A., & Cardoso, J. (2007). What makes process models understandable? In *Business Process Management* (pp. 48-63): Springer.
- Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3), 243-276.
- Moody, D. L. (2009). The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6), 756-779.
- Narciss, S. (2008). Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*, 125-144.
- Nelson, H. J., Poels, G., Genero, M., & Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal*, 20(1), 201-228.
- Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199-218.
- Petre, M. (1995). Why looking isn't always seeing: readership skills and graphical programming. *Communications of the ACM*, 38(6), 33-44.
- Pinggera, J., Stefan, Z., Matthias, W., Dirk, F., Barbara, W., Jan, M., & Reijers, H. A. (2012). Tracing the process of process modeling with modeling phase diagrams. In *Business Process Management Workshops* (pp. 370-382): Springer Berlin Heidelberg.
- Poelmans, S., & Wessa, P. (2013). A Constructivist Approach in an e-Learning Environment for Statistics. *Interactive Learning Environments (accepted)*.
- Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., & Xue, N. (2011). CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task* (pp. 1-27). Portland, Oregon: Association for Computational Linguistics.
- Recker, J., Safrudin, N., & Rosemann, M. (2010). How novices model business processes. In *Business process management* (pp. 29-44): Springer.
- Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(6), 601-618.

- Santos, J., Govaerts, S., Verbert, K., Duval, E. (2012). Goal-oriented visualizations of activity tracking: a case study with engineering students. In *2nd Conference on Learning Analytics and Knowledge* (pp. 143-152).
- Schenk, K. D., Vitalari, N. P., & Davis, K. S. (1998). Differences between Novice and Expert Systems Analysts: What Do We Know and What Do We Do? *Journal of Management Information Systems*, 15(1), 9-50.
- Sedrakyan, G., Snoeck, M., & Poelmans, S. (2014). Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. *Computers & Education*, 367-382, 367 - 382.
- Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1), 153-189.
- Siemens, G., & Baker, R. (2012). Learning analytics and educational data mining: Towards communication and collaboration. *Proceedings of the 2nd Conference on Learning Analytics and Knowledge*, 252-254.
- Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *Educause Review*, 46(5).
- Sikora, E., Bastian, T., & Pohl, K. (2011). *Requirements engineering for embedded systems: An investigation of industry needs.*: Springer Berlin Heidelberg.
- Snoeck, M. (2014). *Enterprise Information Systems Engineering: The MERODE approach*: Springer.
- Snoeck, M., Michiels, C., & Dedene, G. (2003). Consistency by construction: The case of MERODE. In M. A. Jeusfeld, Pastor, Oscar (Ed.), *Conceptual modeling for novel application domains, LNCS* (Vol. 2814, pp. 105-117).
- Szlenk, M. (2006). Formal semantics and reasoning about uml class diagram. In *Proceedings of the International Conference on Dependability of Computer Systems, 2006. DepCos-RELCOMEX'06.* (pp. 51-59): IEEE.
- van der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., & Weijters, T. (2009). ProM: The Process Mining Toolkit. *BPM (Demos)*(489).
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3).
- Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. *Decision Support Systems*, 15(4), 285-304.
- Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning Subjective Language. *Computational Linguistics*, 30(3), 277-308.
- Zimmerman, B. J. (2008). Investigating Self-Regulation and Motivation: Historical Background, Methodological Developments, and Future Prospects. *American Educational Research Journal*, 45(1), 166-183.

Appendix 1

KU Leuven exam supervision system

To optimize its exam supervision system the Faculty of Business and Economics decided to build a web interface which should allow automating certain tasks of the exam supervision assignment and monitoring process. After implementing a first pilot version of the system at the level of the faculty, the system was demonstrated to the administrative directors of the other faculties of KU Leuven. It was decided that the pilot version should be expanded to a university-wide system that can be used yearly. The system should satisfy the following particular requirements:

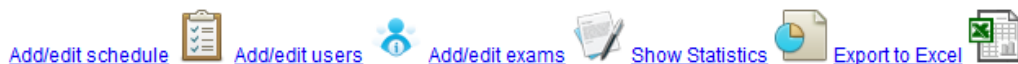
The system is implemented in a university-wide manner: this means the system is not replicated for each faculty separately, but collects all data into a single system. Nevertheless, people and students within a faculty should view only the data that is relevant for that faculty as if the system had been made for them only. Likewise, the system will collect data year after year, but within an academic year, for operational views, one should only view the data of that academic year.

Permissions for KU Leuven employees (professors, administrative staff and research/teaching assistants) are managed according to the departments people belong to. If a person belongs to two different departments (e.g. because combining part time assignments), this person will have access to the data of the different faculties these departments belong to.

In order to filter data per faculty and academic year, the system relies on the way course ownership and teaching assignments are managed. The program book consists of a list of educational programs. Each educational program is "owned" by a "Permanent Educational Commission" (or PEC for short), that manages the program. Each PEC belongs to exactly one faculty. As courses can be shared across programs, for each course, it is decided which PEC owns the course. This ownership can be modified across years. In a similar way, professors are assigned to courses on a yearly basis. *(Note that for this case, the study programs a course is part of is out of scope for this system: only the owning PEC is required information).*

There are three exam periods in each academic year taking place in January, June and August. For each season the exams' schedule should be available via the web interface at least a week prior to an exam period for supervisors to view their supervision assignments: exams to be supervised, location (room), date, start and end time, professor(s) names teaching the course(s). Supervisors can have one of two roles: lead supervisor or ordinary supervisor, the meaning of which is that only "lead" supervisors are responsible to contact the professor(s) of an exam moment (one or more exams taking place in a single room) to receive exam copies and any specific instructions to distribute among other supervisors of that exam (e.g. closed-book exam or an open-book, written or submitted online, etc.). For each exam moment, one of the supervisors is nominated lead supervisor. These supervisions are assigned by the people from the student office who can additionally modify the schedule by adding/removing/modifying exam moments, supervisors, their roles, etc. The student office should be able to make this administration through an "admin view" accessible only to them. The student office is only responsible for managing the exams and supervisions of courses that are owned by a PEC of their own faculty. The information about exams for courses of other faculties can be consulted, but not managed.

Below is a screenshot of admin view features:



When the schedule is ready supervisors are informed about their supervision assignments by automatically generated e-mails:

Update: Exam Supervision Platform

webmaster@econ.kuleuven.be

Sent: Fri 21/12/2012 09:32

To: Annette Schminke

Cc: Gayane Sedrakyan

You have an update for an exam supervision:

Date	Start time	End time	Exam	Room	Invigilator
15/01/2013	09:00:00	12:00:00	G0Q24A Statistics for finance and insurance (Prof. Tim Verdonck)	HOG 02.28 (UCL Students)	Schminke Annette

Supervisors can however make switch requests (e.g. if they are not available for the specified date). This can be done through the “schedule view” accessible to supervisors.

Below is a screenshot of the supervisors’ user interface:

Exam Supervision Platform										
The user is authenticated Welcome "ECONn11013"										
Search by <input type="text" value="name"/> as <input type="text"/> <input type="button" value="Filter"/>										
Mail to Admin										
date	starttime	endtime	exams	room	name	lead	switch	comments	e-mail	
1	16-08-2013	09:00:00	13:00:00	1) APPLIED MARKETING RESEARCH (TEW) 2) FINANCIAL ENGINEERING (1) Dewitte; 2) Leoni	Pc klas HOG (8 + 1)	Nijs Sanne (78925)				Edit
2	16-08-2013	09:00:00	13:00:00	1) APPLIED MARKETING RESEARCH (TEW) 2) FINANCIAL ENGINEERING (1) Dewitte; 2) Leoni	Pc klas HOG (8 + 1)	Goddyn Hannelore (43753)				Edit
3	16-08-2013	09:00:00	13:00:00	1) ENVIRONMENTAL-TRANSPORTATION ECONOM (7) 2) Welfare, Inequality and Poverty (1) 3) Advanced Industrial Economics (3) 4) Currency Derivatives Markets (1) 5) Actuarial and Financial Models (3) (1) Proost/Van der Loo; 2) Decoster/De Blander; 3) Verboven; 4) Seru; 5) J.Dhaene	AUD.J.VAN DER EECKEN (7 + 1 + 3 + 1 + 3)	Vervenne Jan-Bart (47498)				Edit
4	16-08-2013	09:00:00	13:00:00	1) ENVIRONMENTAL-TRANSPORTATION ECONOM (7) 2) Welfare, Inequality and Poverty (1) 3) Advanced Industrial Economics (3) 4) Currency Derivatives Markets (1) 5) Actuarial and Financial Models (3) (1) Proost/Van der Loo; 2) Decoster/De Blander; 3) Verboven; 4) Seru; 5) J.Dhaene	AUD.J.VAN DER EECKEN (7 + 1 + 3 + 1 + 3)	Backiel Aimée (86305)				Edit
5	16-08-2013	09:00:00	13:00:00	WISKUNDE VOOR ECONOMEN (Quaegebeur/Nicaise)	AULA PIETER DE SOMER (255)	Baert Lieslot (85626)				Edit
6	16-08-2013	09:00:00	13:00:00	WISKUNDE VOOR ECONOMEN (Quaegebeur/Nicaise)	AULA PIETER DE SOMER (255)	Sanchez Alfaro Maria Cecilia (76065)				Edit
7	16-08-2013	09:00:00	13:00:00	WISKUNDE VOOR ECONOMEN (Quaegebeur/Nicaise)	AULA PIETER DE SOMER (255)	Bardoutsos Anastasios (86429)				Edit

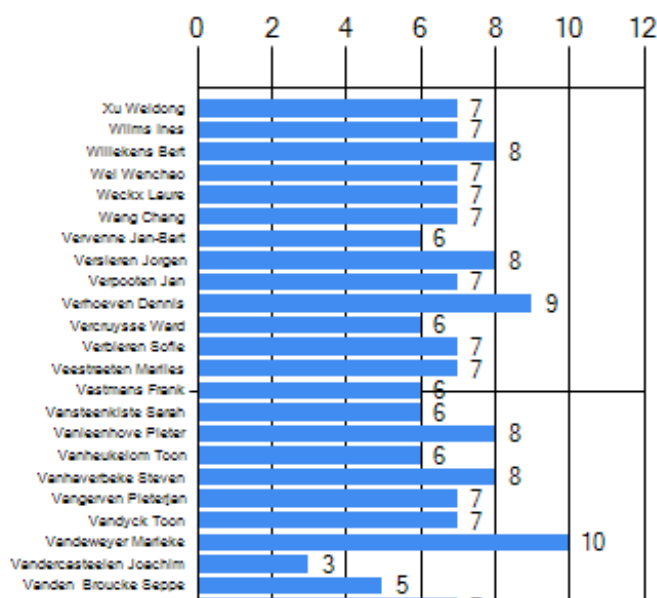
Professors can request a view of the exams’ list for the courses taught by them to see the supervisors list for those exams. Professors can teach several courses (at different faculties) and therefore can be associated with several courses in the system. A course, in turn, can be taught by several professors. A course taught by more than one professor is still managed as a single course, though: it has only one course description and one exam. In order to ensure a smooth administration, professors need to manage the information in their course syllabi carefully. However, not all information can be changed at any time: there are several degrees of modifiability. Between January 15th and March 15th of the preceding academic year, professors can update any information. Importantly, the evaluation type (oral or written) and the number of students they wish to interrogate during 1 exam (in case of oral exam) need to be specified. Depending on this information, one or more exams will be scheduled for the course. If the exam is in a written form supervision will be organized. Professors should also specify the type of an exam (e.g. closed book, open book) for the exams with similar requirements to be grouped in exam moments by the administrators of the system. Likewise they can specify the type of questions (Multiple Choice, open) and duration of the exam. The latter type of information can be modified at any time until the start of the registration via ISP system (around September 15th). Furthermore professors or the student office can cancel a scheduled exam at any time before the opening of registration via the ISP-system. After that

date, they can only cancel an exam if no student has booked this exam in his/her IER (see further down). In general, after a course has been opened for registration, information related to that course can only be ended (*information is never really deleted*) after the academic year has been closed.

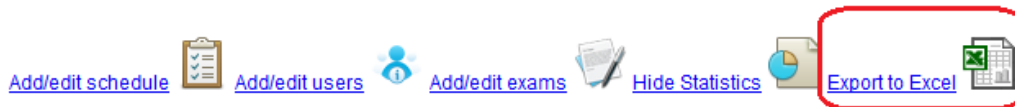
Each course will have at least one final exam per season. Each such exam can be distributed among several exam moments running simultaneously, e.g. the same exam taking place in different rooms (e.g. 300 students will participate in exam X which due to the room availability or capacity limitations will take place in room A, room B and room C each with a capacity of 100 students). Exams for small groups of students are typically grouped into one room to ensure the efficient use of room and supervisor capacity. Therefore, one exam moment can comprise several exams (e.g. exam moment A comprises exams X, Y, Z).

Each exam moment should be assigned one or more supervisors depending on the number of participating students. It is therefore necessary to know per exam moment, not only which courses' exams were grouped, but also which students are booked for that exam in that room. This means that the information is linked to the "individual exam roster" (IER) of each student. Students have an "Individual Study Program" (ISP) for each program they are subscribed to. This ISP contains a course booking for each course the student intends to follow and to take exam of. Each of this bookings is associated with one scheduled exam for that course, hereby constituting the IER of the student. (*Note: the processes on ISP and IER composition and approval are out of scope for this case.*)

Supervisors can supervise several exam moments per exam period. The number of supervisions per exam period on average is 3 per "research assistant" supervisor and 5-6 per "teaching assistant" supervisor. However they can switch exams between exam periods too, e.g. a supervisor can unconditionally accept a supervision of another supervisor which will increment his/her supervisions' number. Normally this results in being assigned one supervision less in the next exam supervision period, and the supervisor whose supervision has been unconditionally accepted without a switch agreement can be assigned one supervision more the next period. To have a balanced assignment of supervisions, the student office can consult the statistics view to monitor the total number per supervisor. Total numbers are calculated as a sum of the supervisions per supervisor (evening and weekend supervisions are counted as double).



Each view can be exported to excel:

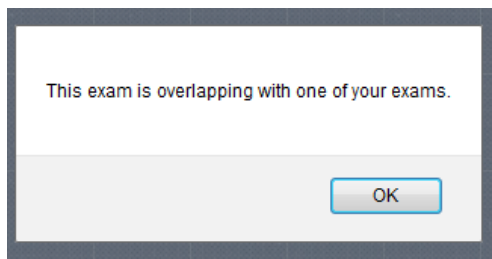


The switch process: a supervisor can set a switch request flag, meaning that s/he is willing to find a replacement for that supervision or switch with someone else's supervision. S/he can also leave some message in a "comments" field for others to see his/her availability, e.g. "any other day would be OK for me to switch".

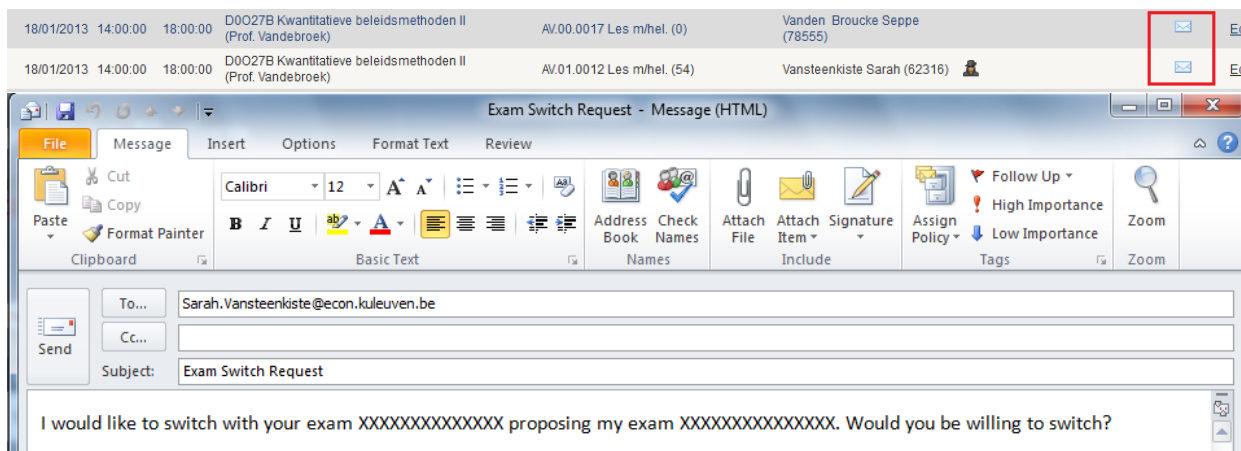
136	31-08-2013	09:00:00	13:00:00	MARKTEN EN PRIJZEN (De Bruyne)	AULA PIETER DE SOMER (270)	Damen Sven (85607)		Edit
137	31-08-2013	09:00:00	13:00:00	MARKTEN EN PRIJZEN (De Bruyne)	AULA PIETER DE SOMER (270)	Willekens Bert (64717)		Edit

To exchange with someone else's exam they can search for switch requests indicated by other users. Once finding appropriate exam moment with a "switch request" flag they can write individual mails to the users with an indication of switch request (red flag in the appropriate cell of the schedule view) to arrange an exchange agreement. Once having made a mutual agreement to exchange turns through mail correspondence, they can simply accept each other's turn (exam moment supervision).

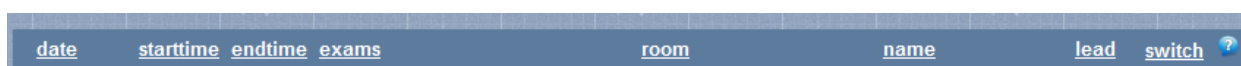
Trying to accept a supervision for which there is an overlapping supervision by the same user should be prevented by the system. The system should also prevent from accepting supervisions the date of which is already passed.



To communicate with each other the rows should be clickable popping up a mail window with a prefilled message template and recipient for the requests to be (modified and) sent.



For ease of use the schedule's columns should be all sortable and filtering should be enabled to search by date, time, supervisor's name, exam name, room number, etc.



The supervision schedule shows the recent updated view on supervisions taking into consideration all the switch request acceptances. If there are multiple switches in the system for a certain exam supervision only the latest successful

acceptance is considered by the information services responsible for the schedule view. However this will not affect the original assignments in the system which will be kept in the system for further reporting purposes (e.g. how many canceled exams, how many switch requests, switch request and acceptance dates...).

So, if Tom requests a switch and Gayane accepts it, and then later on Gayane requests again a switch for the same supervision, and Filip accepts it, then only Filip will show up in the view, but the original supervision by Tom and the switch requests and acceptances by Tom, Gayane and Filip all will be kept in the system.

Appendix 2

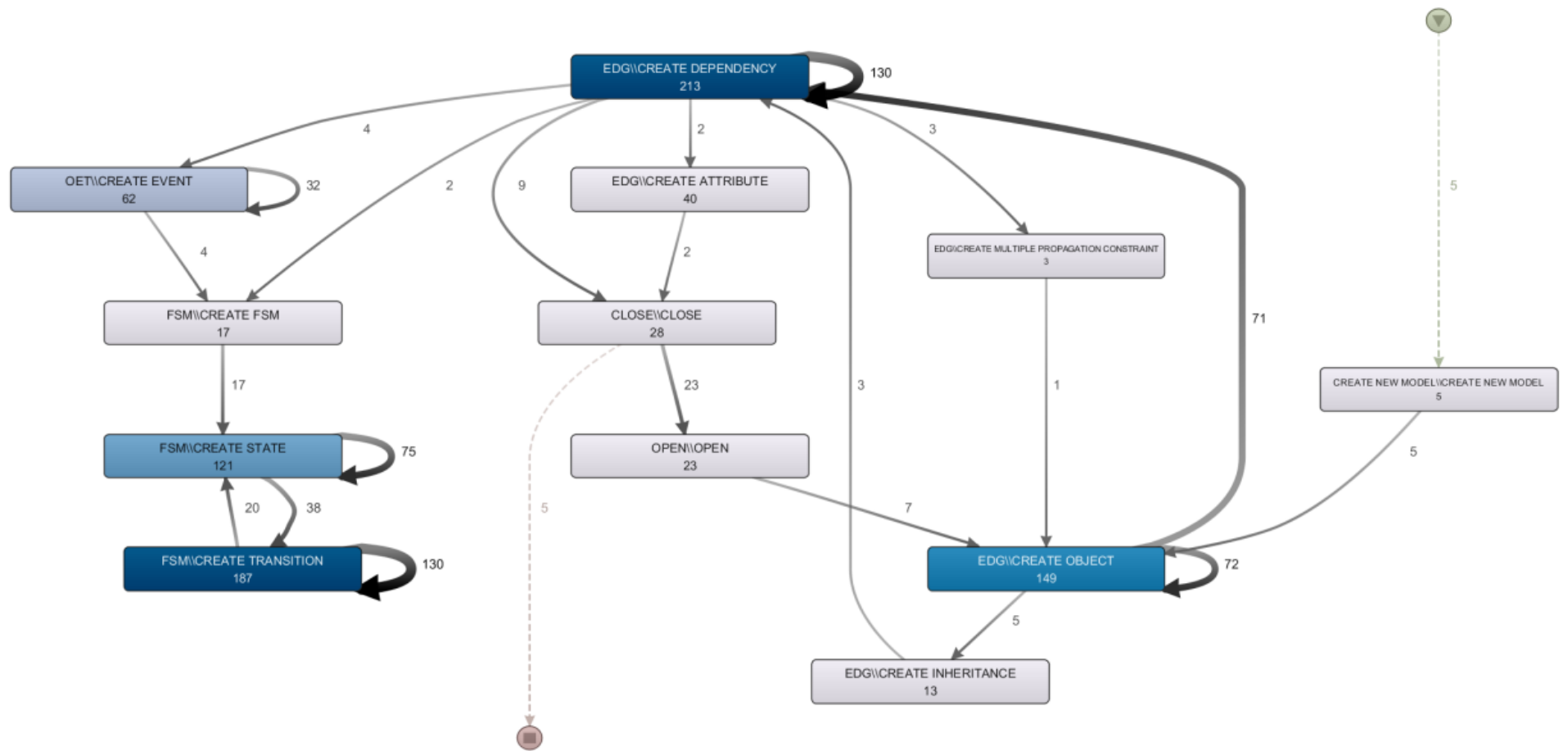


Figure 14. Early sessions modeling Activities - Best 5 cases - Without abstraction

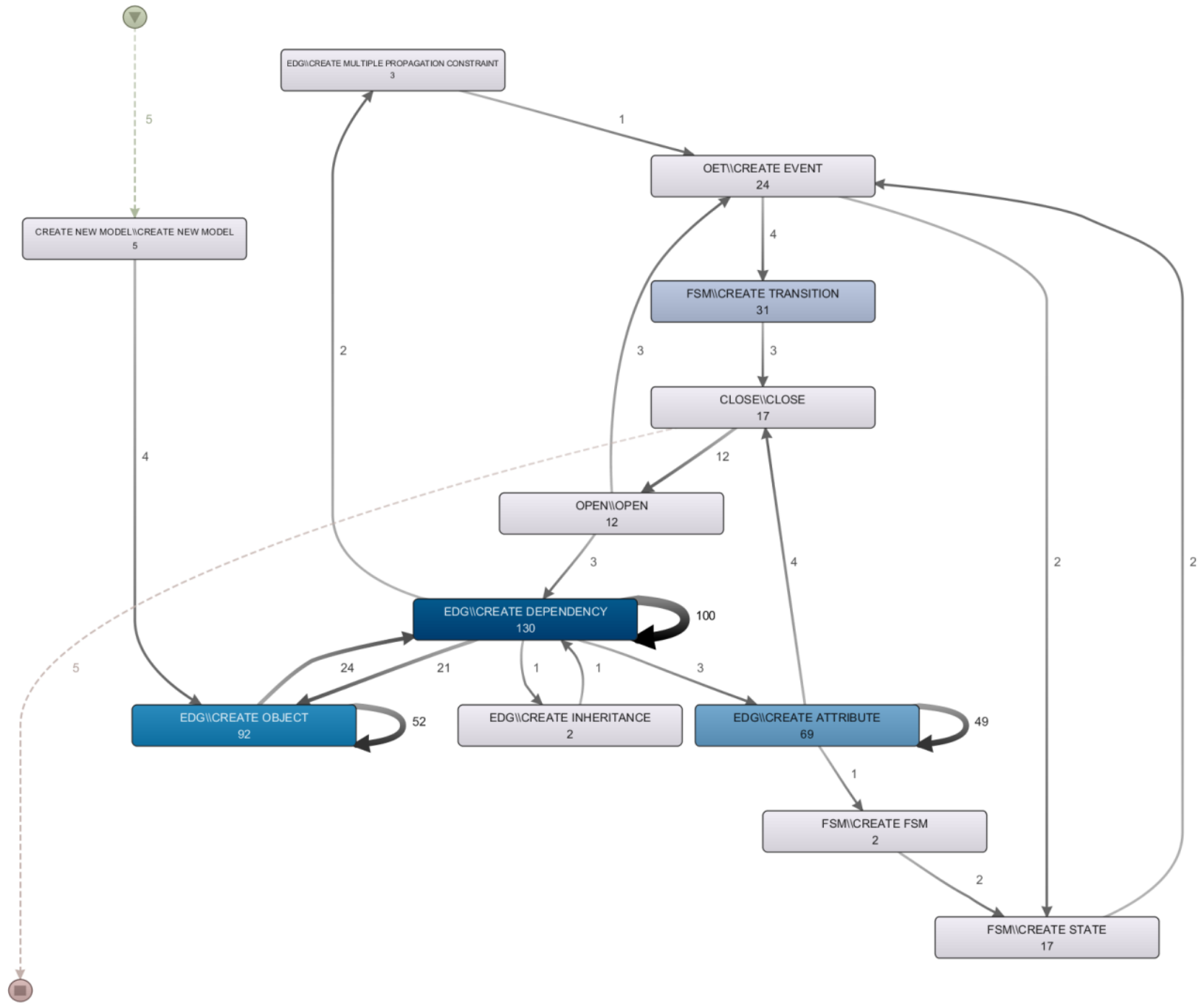


Figure 15. Early sessions modeling Activities - Worst 5 cases - Without abstraction

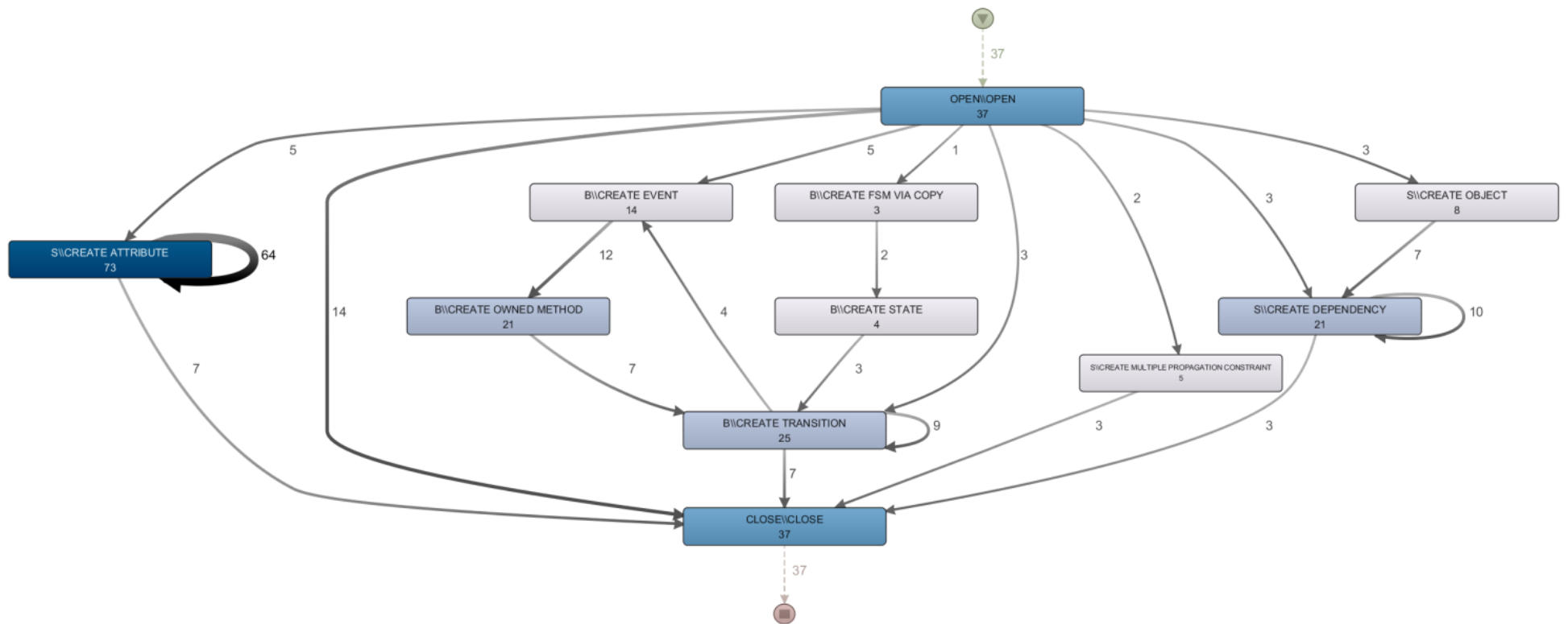


Figure 16. Late sessions modeling Activities - Best 5 cases - Without abstraction

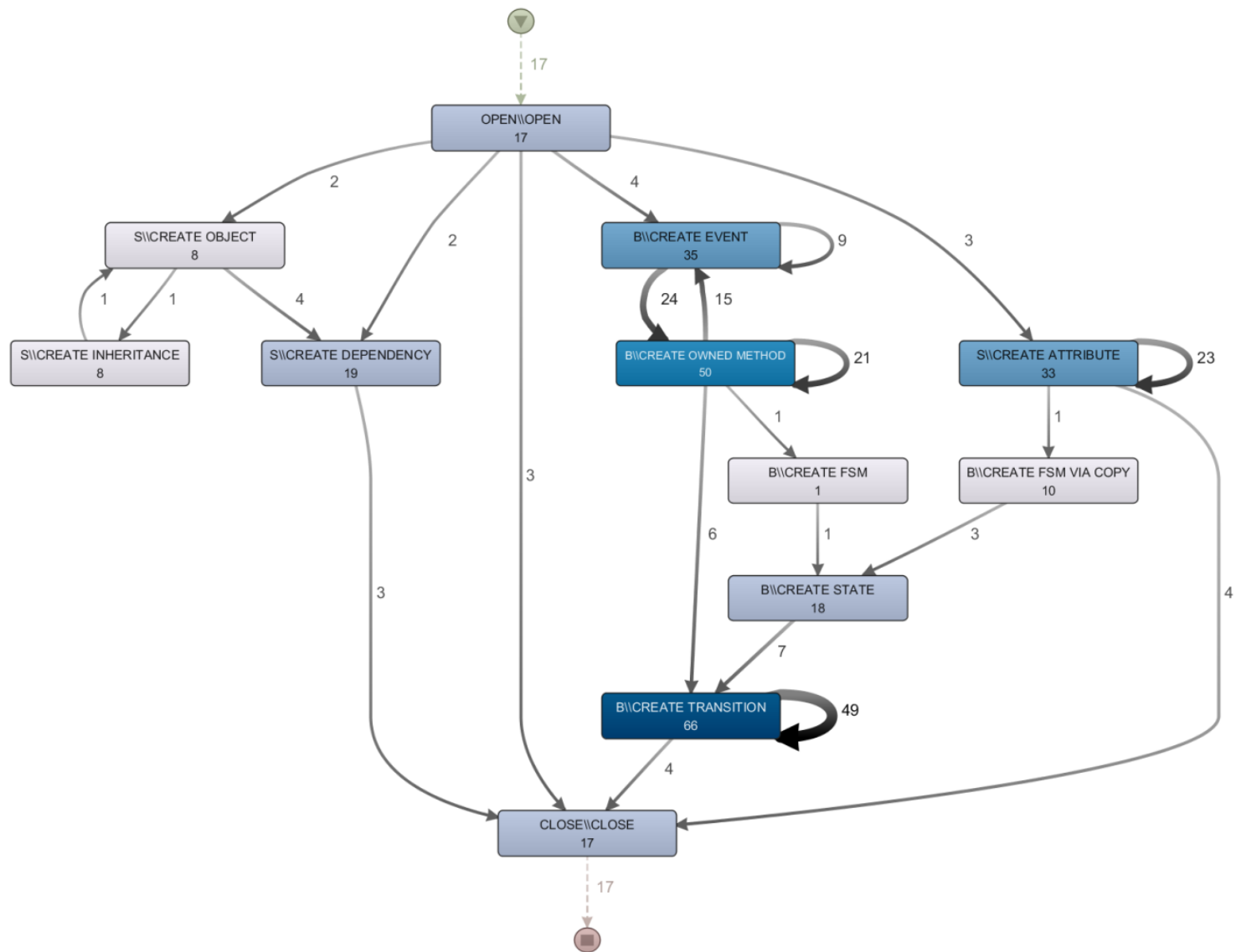


Figure 17. Late sessions modeling Activities – Worst 5 cases - Without abstraction